18th ICCRTS

“C2 in Underdeveloped, Degraded and Denied Operational Environments”

18th ICCRTS

“C2 in Underdeveloped, Degraded and Denied Operational Environments”

Title of Paper:

The Utilization of Network Enabled Capability in NATO Air C2 and Targeting Systems

Topics:

Topic 8: Networks and Networking

Topic 7: Architectures, Technologies, and Tools

Topic 5: Experimentation, Metrics, and Analysis

Authors:

Mutlu Uysal, PhD

NATO Communications and Information Agency

mutlu.uysal@ncia.nato.int

Orhan Cetinkaya, PhD

NATO Communications and Information Agency

orhan.cetinkaya@ncia.nato.int

Point of Contact

Mutlu Uysal, PhD

NATO Communications and Information Agency

The Hague, Netherlands

mutlu.uysal@ncia.nato.int

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **JUN 2013**	2. REPORT TYPE	3. DATES COVERED **00-00-2013 to 00-00-2013**

4. TITLE AND SUBTITLE **The Utilization of Network Enabled Capability in NATO Air C2 and Targeting Systems**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **NATO Communications and Information Agency,The Hague, Netherlands,**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Presented at the 18th International Command & Control Research & Technology Symposium (ICCRTS) held 19-21 June, 2013 in Alexandria, VA. U.S. Government or Federal Rights License

14. ABSTRACT
With the developments in information and network technologies, more systems began to be connected to each other in a network environment and interoperability between them became a crucial requirement. In order to achieve this interoperability, systems should be adapted to more net-centric solutions where the information is shared, rather than stovepipe approaches where the information is for local use only. This paper will focus on the experiences of the NCI Agency Command and Control team during its Network Enabled Capabilities (NEC) studies with the Integrated Command and Control (ICC), Air Command and Control Information Services (AirC2IS) and Joint Targeting System (JTS) and will aim to share the practical lessons learned with the community.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**14**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

The Utilization of Network Enabled Capability in NATO Air C2 and Targeting Systems

ABSTRACT

With the developments in information and network technologies, more systems began to be connected to each other in a network environment and interoperability between them became a crucial requirement. In order to achieve this interoperability, systems should be adapted to more net-centric solutions where the information is shared, rather than stovepipe approaches where the information is for local use only.

This paper will focus on the experiences of the NCI Agency Command and Control team during its Network Enabled Capabilities (NEC) studies with the Integrated Command and Control (ICC), Air Command and Control Information Services (AirC2IS) and Joint Targeting System (JTS) and will aim to share the practical lessons learned with the community.

Keywords: Network Enabled Capability (NEC), Service Oriented Architecture (SOA), web services, Command and Control, Targeting

1. Introduction

The philosophy under "net-centric solutions" or in other words "network-enabled capability" is to provide the right information, at right place at right time to increase the operational effectiveness in a multi-functional and multi-national environment such as a NATO domain. This Network Enabled Capability (NEC) involves the seamless linking of sensors, weapon systems, command and control systems and decision makers in a collaborative environment for planning, assessment and execution. NEC will be composed of a dynamic, networked coalition of military forces, cooperatively sharing information to progressively improve coordination, collaboration and coherency across the full spectrum of military activity [1].

NEC is also recognized as being essential for meeting future challenges in the Trans-Atlantic environment. Through NEC, nations and NATO seek to capitalize on the use of new development approaches to improve operational effectiveness through the networking of their operational capabilities. The development of a NATO NEC (NNEC) is viewed by many nations as the most effective way for their nation to be able to use their own investments to the full in information age technologies in supporting future coalition operations.

In the current state-of-art, this network-enabled capability is achieved by integrating different systems within a Service Oriented Architecture (SOA). SOA is "a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations" [2].

Web services are today's most widely used and popular implementation of service oriented architecture. Note that, SOA is the architectural style, whereas the web services are the current most popular implementation of this architecture. Over the past few years, more systems began to be interoperable with each other using web service technology [3].

Since, NEC is recognized as being essential for meeting future challenges in the Trans-Atlantic environment [1], many NATO systems like Integrated Command and Control (ICC), Air Command and Control Information Services (AirC2IS) and Joint Targeting System (JTS) utilize this concept during their development. Considerable work has been performed and currently being performed to apply the NEC concept to the aforementioned systems to improve the operational effectiveness.

All of these studies showed that NEC concepts offer many benefits by providing a natural decoupling between the services and the clients of those services. This decoupling results in fewer dependencies between systems, enabling systems operating on different budgets, timeframes or schedules to change without impacting each other. As a result, changes in one application will not force a change in another application, which is also called as loose coupling. Implementing loosely coupled integration approaches reduces the complexity and hence the cost of integrating in distributed computing environments. SOA also helps to move application infrastructure from an inefficient, inflexible model - with vertical, stovepipe applications - to a less expensive, enterprise-wide model that delivers a reusable suite of interoperable services.

Although NEC provides many benefits for systems, it also introduces many challenges as well. Quality of the services is one of the most important challenges that should be handled. In a complex network environment, it may not always be possible to get the same quality of service to achieve the operational effectiveness. Consequently, there should be mechanisms to ensure that quality of the service is at an acceptable level even at degraded and denied operational environments and it should be continually monitored. Security is another issue that should also be handled. Services are for sharing the data but it must be noted that the data should be shared with trusted parties only and not to every system in the network. There should be enterprise level security mechanisms which ensure that data is securely shared among the providers and consumers.

The objective of this paper is to share the experiences and lessons learned of NATO C2 team regarding to ICC, JTS and AirC2IS projects through applying the NEC concepts. The paper will start with an introduction section in which basic information about NNEC concepts and above mentioned NATO systems is provided. Then, technical details about how these systems utilize the NNEC concepts will be explained. After that, lessons learned and challenges faced when applying SOA concepts will be discussed. Finally, some alternative approaches to overcome the current challenges will be provided as a future work. We believe that outcome of this paper may also be helpful for other NATO and national command and control systems, which are willing to exchange data using this state of art technologies.

2. Background

2.1 Web Services Basics

A web service is a collection of protocols and standards which enables integration between heterogeneous systems. Web services provide a standard means of interoperability between different software applications, running on a variety of platforms and/or frameworks. The most common web services are XML-based information exchange systems using a Web Services Description Language (WSDL) to describe its services [3].

The typical usage of web services is synchronous request-response, where clients simply make a request and get the response from the server. For most systems, this approach is sufficient to implement and interface with other systems. The service registry component may also be used to publish the service interfaces by the producers and to search the available services by the consumers (**Figure 1**).

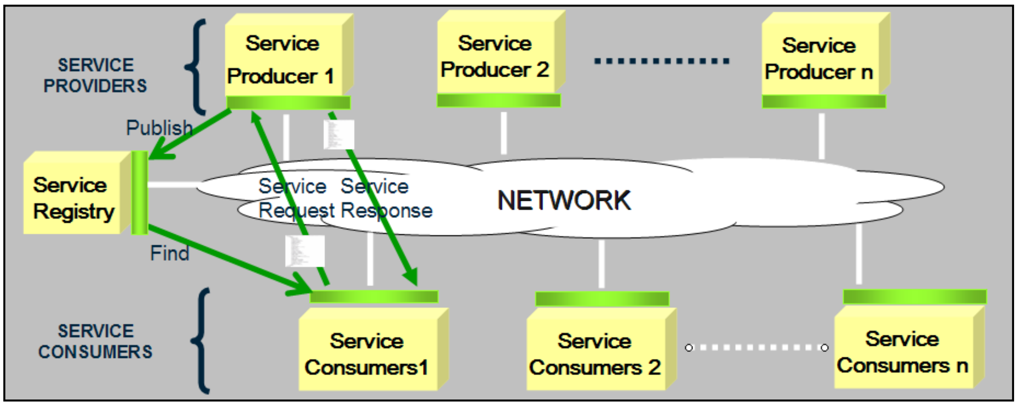

Figure 1: Service Oriented Architecture using web services

2.2 Integrated Command and Control (ICC)

ICC is an Integrated Command, Control, Communications and Intelligence (C3I) environment that provides information management and decision support to NATO Combined Air Operations Centre (CAOC) level air operation activities during peacetime, exercise and wartime. The ICC provides functional support for the most critical Air C2 functions at the CAOC level, such as Planning and Tasking, Air Task Order (ATO)/Air Task Message (ATM) generation, Airspace Control Orders, and Current Operations (Defensive and Offensive section) (**Figure 2**).

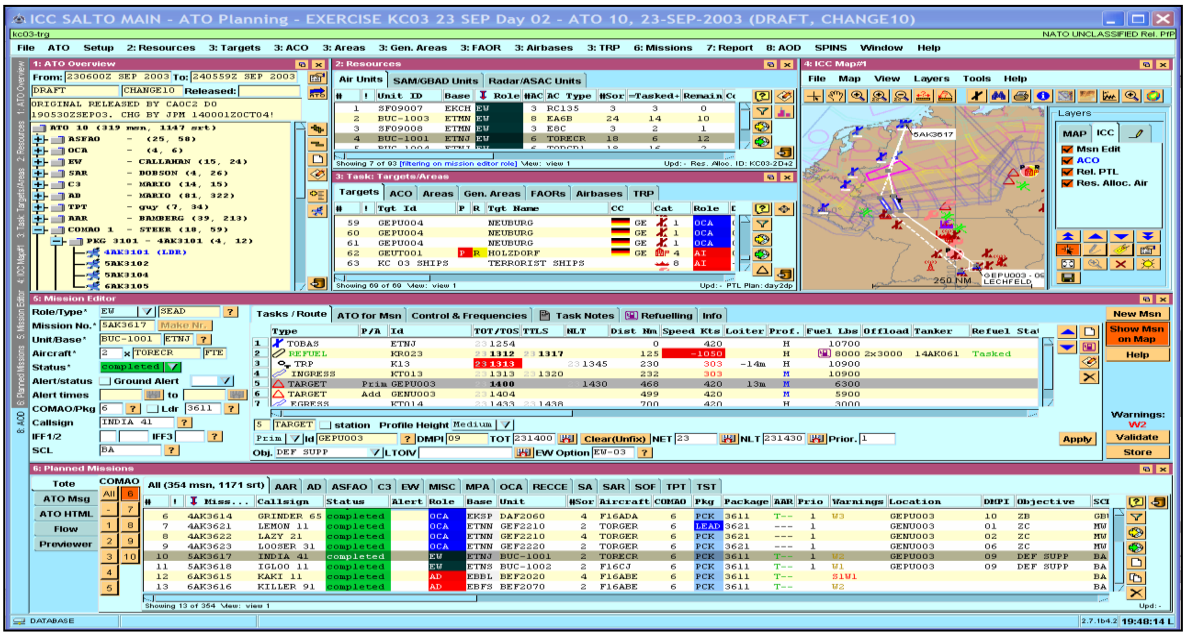

Figure 2: Screen-shot from ICC

The standard ICC architecture is a "3-tier architecture" which includes a database server based on an Oracle database, a COSI layer as the middle-tier (containing business logic) and finally a client application running on the desktop. ICC clients open a CORBA connection to the COSI layer to perform get/set operations. In the enhanced version of ICC for NEC, a web service layer, called WISI, is added on top of the COSI layer (**Figure 3**). With the help of this additional layer, it becomes easier to reach ICC data for all other 3rd party applications using standard protocols such as HTTP, SOAP and XML. This new approach and capability provides a new mechanism for other systems to interoperate with ICC.

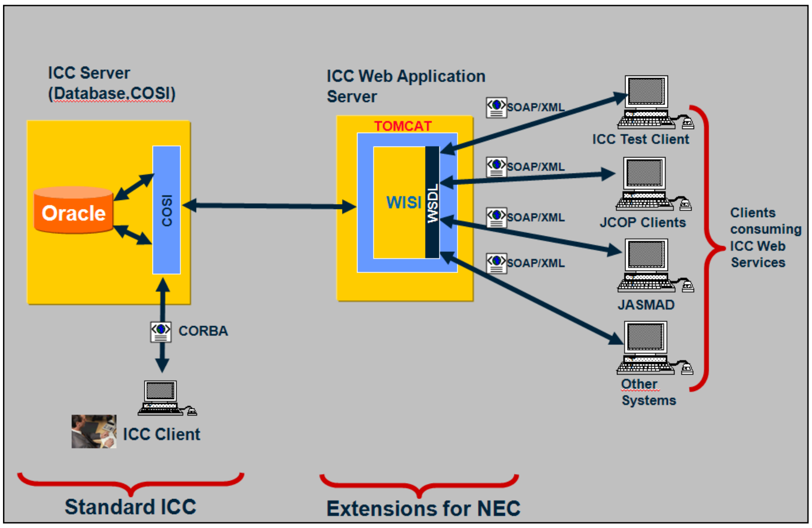

Figure 3: ICC web service architecture

2.3 Joint Targeting System (JTS)

Joint Targeting System (JTS) is an automated tool layer supporting NATO joint targeting process which is essentially a cyclic process that begins with the Joint Force Commander’s (JFC's) guidance and objectives and sequentially steps through target development including target list management, weaponeering assessment, force applicable execution planning and mission execution, and combat assessment. JTS architecture has many similarities with ICC; therefore, it is “3-tier architecture” with additional web service layer supporting NNEC (**Figure 4**).

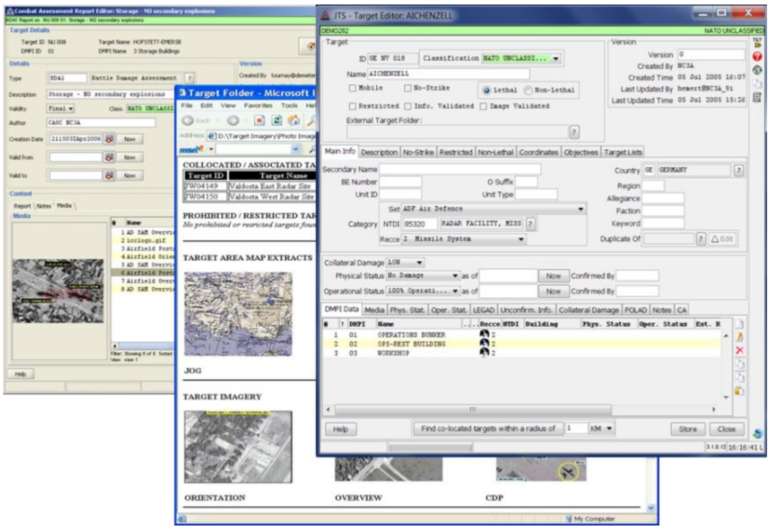

Figure 4: Screen-shot from JTS

2.4 Air Command & Control Information Services (AirC2IS)

NATO Air Command & Control Information Services (AirC2IS) is a strategic and operational level command and control information system which provides an automated capability for supporting NATO operational staff to continuously adapt to the constantly changing NATO environment and to address the security challenges. AirC2IS will support the joint air planning, tasking, monitoring, and analysis efforts for NATO air operations, including Tactical Ballistic Missile Defence (TBMD) operations.

AirC2IS leverages a modern, robust, integrated and flexible Service Oriented Architecture (SOA)-based solution. It is the first example of the NATO Network Enabled Capability (NNEC) from the design, delivering the first air and TBMD service libraries to the NNEC services framework.

Integration capability provides flexibility to operate with other Bi-Strategic Command Automated Information Systems (Bi-SC AIS) components and national systems. AirC2IS interacts with joint, land, maritime, air and other information systems to support operational tasks. It will utilize existing Bi-SC AIS core services and other NATO Command and Control (C2) applications such as Core Enterprise Services (CES) to provide an integrated command and control capability to NATO air staff. AirC2IS provides execution of air C2 across all levels of command as a single homogeneous process based on seamless, transparent and timely flow of data and information.

The AirC2IS architecture is designed by taking into consideration newly introduced NATO CES framework and SOA governance. It is anticipated that after the delivery of AirC2IS, several services provided by this architecture will be seen as the first occurrence of the NATO CES envisaged capabilities. This is an important step to achieve service standardization and improve interoperability within the Bi-SC AIS environment (NATO enterprise).

AirC2IS utilizes a layered architecture approach to support complex operational requirements with good maintainability, reusability, scalability, strength and security which is depicted at the **Figure 5**:

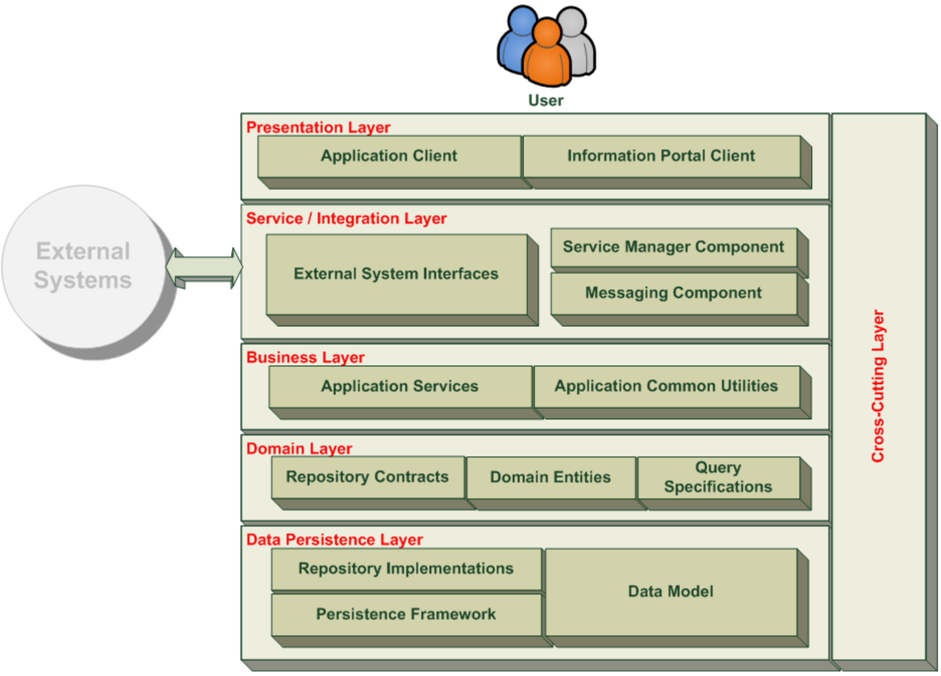

Figure 5: AirC2IS layered software architecture

The layers of AirC2IS are defined as:

- Presentation Layer provides application's user interfaces.
- Service/Integration Layer provides access to all the services and the external system information.
- Business Layer provides the business logic/functionality of the application.
- Domain Layer provides visibility to the domain concepts, business processes and domain rules.
- Data Persistence Layer provides the interaction with the databases.
- Cross-Cutting Layer provides the generic technical capabilities to all layers.

System states (Software States) and operational modes (Technical States) are shown in **Figure 6**.

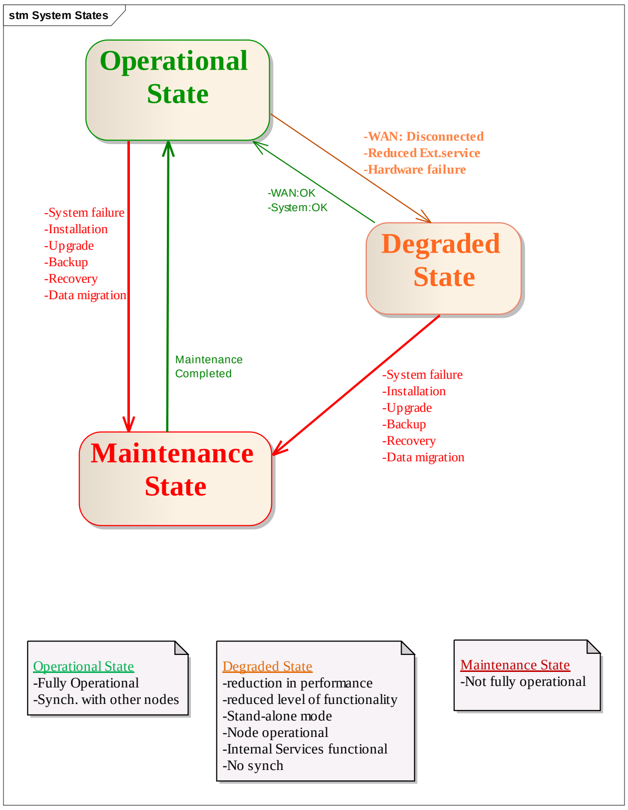

Figure 6: AirC2IS system states and modes

Operational State: In this state, services in the LAN/WAN environment are fully operative. AirC2IS is fully operational and performs synchronisation with other nodes. The operational state will include several HQs at multiple command levels working on a distributed collaborative AirC2 cycle using the NS WAN.

Degraded State: The degraded state will include provision of hardware failure, WAN disconnected and reduced level of service of NS/mission-specific WAN/LAN. In this state AirC2IS will be capable of providing the service to individual Users with some reduction in performance or one or more of the AirC2IS functional services (or Application and Interface Products) may be impacted. Non-automated processes may replace the AirC2IS system services to achieve the system goals. Each AirC2IS node will have the flexibility to fully operate in a disconnected environment (stand-alone mode), without connectivity with other nodes (WAN connection lost). When the proper connectivity between nodes (WAN) becomes available, AirC2IS will change state to Operational State and perform resynchronization.

Maintenance State: In case of system failure or to perform upkeep operations, the system state will be changed to Maintenance State by system administrator. AirC2IS Maintenance state allows for essential maintenance and upkeep operations, including installation, back-up, recovery, upgrade, data migration, and maintenance/repair.

3. The NEC Study Examples

In this section, the ICC, AirC2IS and JTS NEC studies will be explored. Basically, there are four sets of studies that will be focused at this paper:

- ICC in Joint Common Operational Picture (JCOP)
- ICC/JASMAD Airspace Coordination Order (ACO) Exchange
- JTS/ACCS Target and Target Lists Exchange
- ICC/ACCS/TBMCS Mission Exchange

3.1 ICC-JCOP

JCOP aims to provide a common operational picture to the NATO users to increase the situational awareness by collecting information from different systems and aggregating them into a single view. For this purpose, a set of NATO systems such as ICC, Land C2 System (LC2IS), Maritime C2 System (MC2IS), iGeoSIT and NIRIS provide their data using web services. The JCOP uses ICC as a "rich viewer", iGeoSIT as a "light viewer" and JWEB as a "web viewer". The viewers consume web services and display the data on their maps as a joint common operational picture. **Figure 7** shows the consumption of ICC web services by other JCOP viewers as well as the consumption of other systems' web services by ICC.

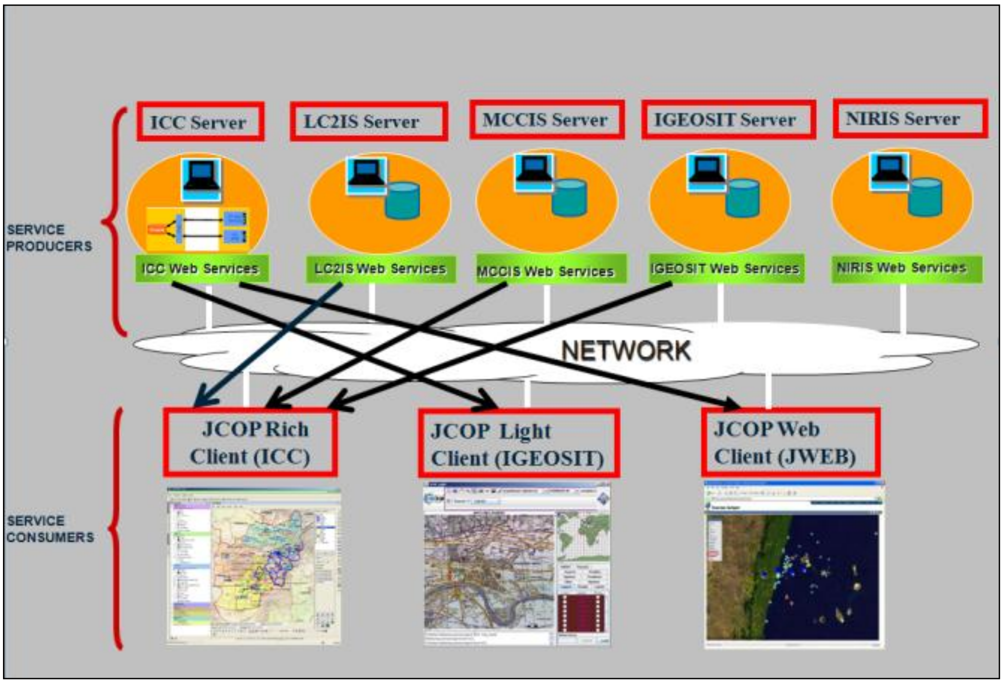

Figure 7: ICC in action in JCOP as both service producer and consumer

Although all the service producers provide their data using web services, the underlying methodology behind the web services are all different. ICC web services are standard XML-based services defined by WSDL. iGeoSIT services are Web Feature Services (WFS) and Web Map Services (WMS) defined by the Open Geospatial Consortium specifications. LC2IS, MC2IS and NIRIS provide web services whose data is in NATO Vector Graphics (NVG) format and also semantic web services using Resource Description Framework (RDF) defined by W3C.

3.2 ICC-JASMAD ACO Exchange

In this experiment, US Joint Air Space Management and Deconfliction (JASMAD) consumed the ICC airspace data through the web services and displayed them on its map. **Figure 8** shows both ICC and JASMAD showing the same ACO.

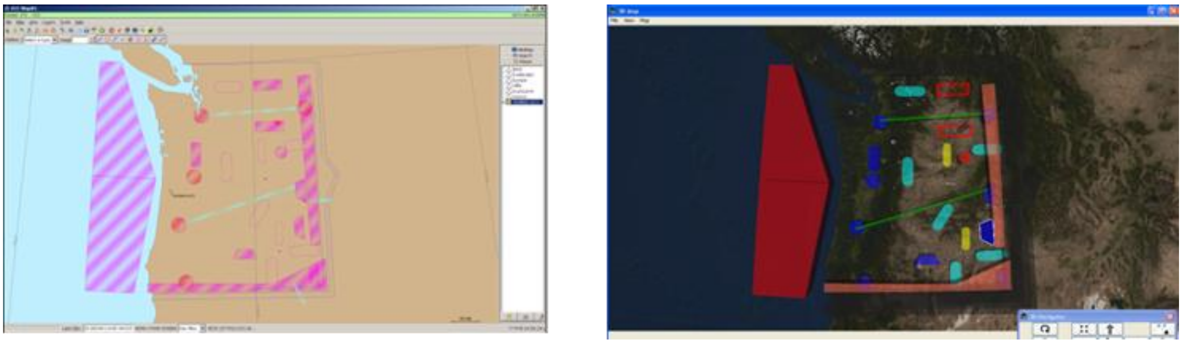

Figure 8: ICC and JASMAD showing the same Air space Coordination Order

3.3 JTS-ACCS Target and Target List Exchange

The aim of JTS-ACCS data exchange experiments was to provide target/target list data to NATO Air Command and Control System (ACCS) while exploring the usage of web services and Oracle’s Service Oriented Architecture (SOA) tools. For this purpose, part of the Prioritised Target List (PTL) and target data was provided to ACCS as web services by WISI. These experimental web services were developed as JTS add-ons to WISI. The Oracle BPEL tool was used to read these web services and pass the results to ACCS after performing the necessary mapping and transformation steps (**Figure 9**).

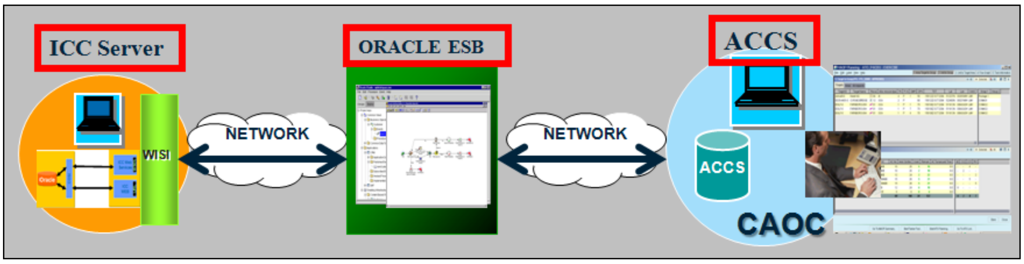

Figure 9: WISI consumed by Oracle’s SOA tools

3.4 ICC-ACCS-TBMCS Mission Exchange

The aim of this experiment was to exchange mission information in a common format between three command and control systems: ICC, ACCS and US Tactical Battle Management Core Systems (TBMCS) using a common methodology. For this purpose, initially, a common mission definition (CMD) was defined which had the same meaning for all of the systems. Then, all the systems defined web service end-points to provide this data to the other systems. Real Simple Syndication (RSS) feeds were used to provide a list of links to a web service end-point per mission. With this approach, the services became also consumable by standard RSS aggregators as well (**Figure 10**).

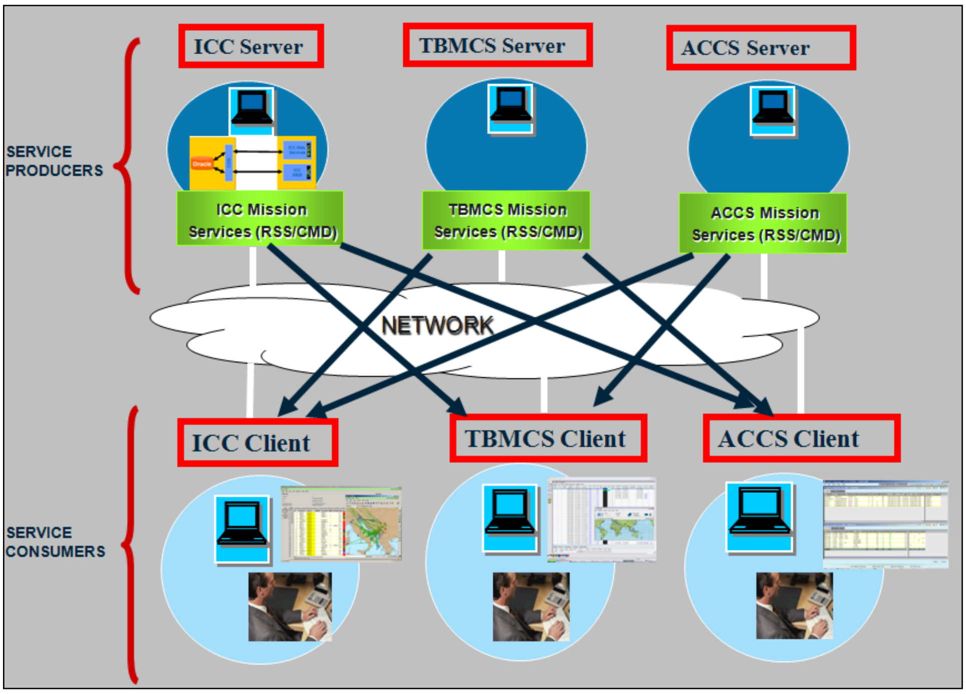

Figure 10: ICC/ACCS/TBMCS exchanging mission data using RSS feeds and web services

4. Lessons Learned by NEC Studies

Many valuable lessons were learned from each of these studies. This section summarizes some of these learned lessons and some challenges that are faced during these studies.

4.1 Towards Service Oriented Architecture

In the past, stovepipe systems were built, which performed their functions very well but were not interoperable with each other. There was a barrier in between the systems running at different domains such as air, land and maritime and even in between the different systems in the same domain. Over time we learned that, we should move our application infrastructure from an inefficient, inflexible model - with vertical, stovepipe applications - to a less expensive, enterprise-wide model that delivers a reusable suite of interoperable services as also shown in **Figure 11**.

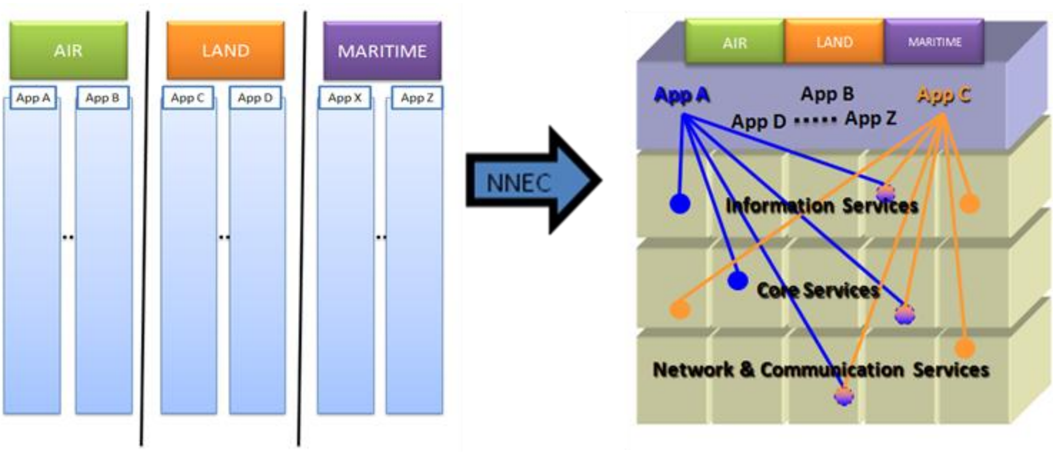

Figure 11: Shift from standalone functional stovepipes to interoperable service oriented systems

This shift to NEC concepts offers many benefits by providing a natural decoupling between the services and the clients using these services. This decoupling results in fewer dependencies between systems. As a result of this, systems operating on different budgets, timeframes or schedules, can be changed without impacting each other. In other words, changes in one application will not force a change in another, which is also called "loose coupling". Implementing loosely coupled integration approaches reduces the complexity and therefore the cost of integrating different heterogeneous systems. This enables us to build modular applications with more flexibility.

In such complex environments, the ESB represents the piece of software that lies between the business applications and enables communication among them. Ideally, the ESB should be able to replace all direct contact with the applications on the bus, so that all communication takes place via the ESB. An ESB brings flow-related concepts such as transformation and routing to a Service-Oriented Architecture. An ESB can also provide an abstraction for endpoints. This promotes flexibility in the transport layer and enables loose coupling and easy connection between services. Using ESB's as a central bridge makes it easy to add new services, change the available ones and manage all the traffic on an on-going basis for an acceptable quality of service.

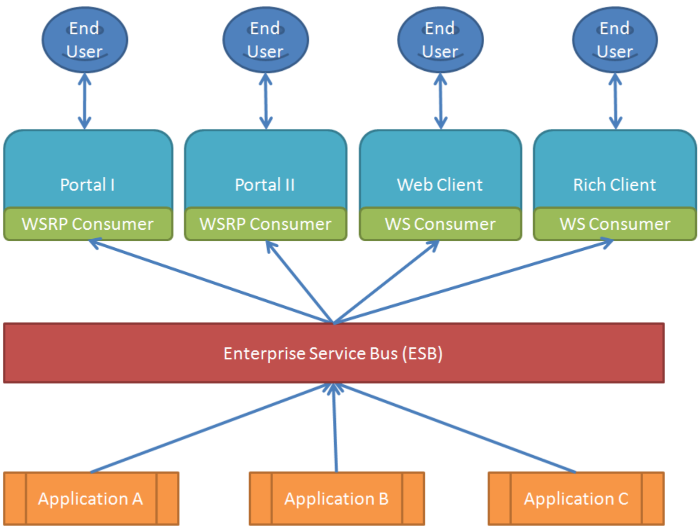

Figure 12: Enhanced Usage of Enterprise Service Buses (ESB) for integration of heterogeneous systems

4.2 Learned Lessons from the NNEC Studies

The learned lessons will be grouped under three sub-headings such as: service definition, quality of service and service provisioning.

Service Definition

- The web service interface should be well-defined to prevent misunderstandings and misinterpretations between the providers and consumers. Service contract must provide unambiguous information about what the service provides. Extra annotations for all defined elements and constraints about all possible data values should be added to the interface descriptions. It is strongly suggested to do this in the schema that is referenced by the web service rather than providing an external document describing the web service interface. By doing so, it has achieved self-documented clear web service interfaces.
- Service granularity is very important in definition of the web service interfaces. Granularity is the extent to which a system is broken down into small parts. Fine-grained services address small units of functionality or data exchange whereas coarse-grained services encapsulate larger

amounts of capability within a single interface. There should be a good balance on the granularity of the services which will result either larger amounts of data to be exchanged in a single call or many service calls each of which is returning a small amount of data. Each of these choices may fit well to different problem domains based on the requirements and the other variables.

- It should be preferred to reference to some commonly accepted models rather than defining new models whenever possible. Referencing to common models makes the job of the consumers easier. On the other hand, referring such a common model may cause loss of some data due to improper mappings between the real data model and the referenced one. Especially for geographical data, it is a good practice to use commonly accepted standards like SVG, GML, KML etc.

Quality of Service

- Quality of service issues should be considered as critical from the start until the end of the projects. Some load balance testing activities should be considered as part of the development process starting from the early phases.
- Applying some compression techniques for web services may help to minimize the bandwidth usage and decrease the transmission time of the XML messages over the network especially in wide area networks.
- Although it is a good practice to use web services for the flexibility that they offer for the interoperability across different platforms and different programming languages, in some cases, there may be need to use the native interfaces for internal tasks for performance matters.

Service Provisioning

- If the number of services available on the network increases, the need for a global service registry becomes vital to ease the configuration and enable dynamic binding between the producers and consumers. Although such a registry was not used in our experiments, NATO has an on-going metadata and service registry project for this purpose, which will be a core component in future.
- Service orchestration helps to build business processes using basic services. There are a lot of commercial and free enterprise service bus tools for this purpose. These tools may be helpful for data transformations and may also provide some core service functionality like security, auditing and monitoring.
- Although the typical usage of web services is synchronous request-response, where clients simply make a request and get the response from the server, there may be cases where asynchronous communication is required such as information dissemination. In these cases, it is advised to use one of the emerging notification standards like WS-Notification or WS-Eventing to meet this requirement.
- Semantic web service technologies promise for future. They are evolving very fast and it is wise to experiment with them. However, we believe that more time is needed to use these technologies in real operational systems because of the following reasons. Firstly, there are not many development tools and experienced developers to support these technologies to be accepted and widely used yet. Secondly, currently available tools supporting these technologies such as Jena, RDFLib etc., needs a learning curve and it may be difficult to use them for both producing and consuming a service. Thirdly, their performance is not as good as classical web services due to extra overhead of RDF structuring. As a result, we believe that, it needs more time to effectively use semantic web services in real operational systems.

4.3 Challenges of SOA

Although NEC provides many benefits for systems, it also introduces many challenges as well. Quality of the services is one of the most important challenges that should be handled. In a complex network environment, it may not always be possible to get the same quality of service to achieve the operational effectiveness. Consequently, there should be mechanisms to ensure that quality of service is at an acceptable level and it should be continually monitored.

Security is another issue that should also be handled. Services are for sharing the data but it must be noted that the data should be shared with trusted parties only and not to every system on the network. There should be enterprise level security mechanisms which ensure that data is securely shared among the providers and consumers. It is advised to refer to some commonly accepted standards such as WS-Security in managing this issue.

One of the other challenges of SOA is that, it requires change in the way the software is traditionally developed. In this new methodology, the focus is reuse rather than re-implement. With this reuse approach, project management becomes more complex due to project level interdependencies. The responsibilities for the reused components should be clearly defined and service level agreements should be available between the providers and consumers. If it requires having an interface change, this should be done in coordination with all the consumers in order not to break the current interfaces. This may even cause to have multiple versions of the same services deployed due to different dependencies.

SOA governance is very critical in order to have a standard approach across the enterprise. It refers to formal policies, processes, and procedures for development and management of services and business processes throughout the SOA lifecycle. Governance is required to define and enforce architectural, technical, and business policies to ensure the promise of SOA is realized [4].

A SOA Governance body is considered a key requirement for implementing a governance model in a SOA environment. Such a body interacts with both developers and end users by defining and addressing areas for governance such as service security, service registry, service lifecycles, service testing. It identifies processes and best practices for on-going development and implementation, and provides the leadership and forum necessary for determining needs, SLAs and dependencies [5].

It should not be forgotten that all the supporting technology for the SOA is still emerging. Although it has proven its success by being used by many systems for years, this is an on-going process and new requirements may appear with new developments in the underlying technology.

5. Summary and Conclusion

The utilization of NEC by NATO systems is recognized as being essential for meeting future challenges and needs in the Trans-Atlantic environment. A lot of investment has been accomplished and considerable work has been achieved in recent years to capitalize on its usage to improve operational effectiveness in NATO. Several experiments/studies have been carried out within many NATO and national C2 systems to utilize this concept in recent years. All of these studies showed that web services offer a key enabler technology to share data and business processes between different systems in a loosely coupled way to achieve NNEC.

In this paper, we focused on the experiences of the NATO Communication and Information Agency Command and Control team from its NEC studies with the ICC, AirC2IS and JTS capabilities and aimed to share the lessons learned with the community. As the requirements and technologies involved are still evolving, our studies on this concept will continue in the future.

6. SELECTED REFERENCES

[1] NATO Network Enabled Capability Feasibility Study (NNEC FS), Volume II, Version 2.0, October 2005, NATO Unclassified

[2] OASIS Reference Model for Service Oriented Architecture 1.0, Committee Specification 1, 2 August 2006

[3] W3C, World Wide Web Consortium, http://www.w3.org/2002/ws/desc/

[4] Greg Bjork (CEO WebLayers), SOA Governance Technical Exchange Meeting, The MITRE Corporation, 18 January 2007.

[5] Jeff Gold, "SOA Governance" Overview Brief, The MITRE Corporation, 5 January 2009.